



全国高等职业教育规划教材

软件测试技术

刘文乐 田秋成 编著

55



电子教案下载网址 www.cmpedu.com



机械工业出版社
CHINA MACHINE PRESS

全国高等职业教育规划教材

软件测试技术

刘文乐 田秋成 编著



机械工业出版社

本书共分为3部分,第1部分为软件测试基础篇,第2部分为软件测试方法篇,第3部分为现代测试技术篇,首先介绍了软件测试的基本概念、测试模型、测试分类及测试的步骤,软件测试设计、测试用例设计、测试管理,接着按照软件测试的分类,依次详细地介绍了生存周期测试方法、黑白灰盒测试方法、功能与性能测试方法,最后介绍了现代测试方法和自动测试方法。

本书不仅注重内容的系统性、科学性,还特别关注测试技术的先进性、新颖性,并在编写时注重理论与实践紧密结合,实用性、可操作性强。

本书不仅可作为高职高专计算机类人才培养的专业教材,又可作为普通大专、本科生的教学、参考用书,还可作为计算机爱好者的自学用书。

本书配套授课电子课件,需要的教师可登录 www.cmpedu.com 免费注册、审核通过后下载,或联系编辑索取(QQ: 1239258369, 电话: 010 - 88379739)。

236402

图书在版编目(CIP)数据

软件测试技术/刘文乐,田秋成编著. —北京:机械工业出版社,2012. 1
全国高等职业教育规划教材
ISBN 978-7-111-36809-0

I. ①软… II. ①刘…②田… III. ①软件-测试-高等职业教育-教材 IV. ①TP311.5

中国版本图书馆 CIP 数据核字(2011)第 261031 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

责任编辑:鹿征

责任印制:乔宇

三河市宏达印刷有限公司印刷

2012 年 1 月第 1 版·第 1 次印刷

184mm × 260mm · 16.5 印张 · 406 千字

0001-3000 册

标准书号: ISBN 978-7-111-36809-0

定价: 32.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

电话服务

网络服务

社服务中心:(010)88361066

门户网:<http://www.cmpbook.com>

销售一部:(010)68326294

教材网:<http://www.cmpedu.com>

销售二部:(010)88379649

读者购书热线:(010)88379203

封面无防伪标均为盗版

全国高等职业教育规划教材计算机专业

编委会成员名单

主任 周智文

副主任 周岳山 林东 王协瑞 张福强

陶书中 龚小勇 王泰 李宏达

赵佩华

委员 (按姓氏笔画排序)

马伟 马林艺 万雅静 万钢

卫振林 王兴宝 王德年 尹敬齐

史宝会 宁蒙 刘本军 刘新强

刘瑞新 余先锋 张洪斌 张超

李强 杨莉 杨云 罗幼平

贺平 赵国玲 赵增敏 赵海兰

钮文良 胡国胜 秦学礼 贾永江

徐立新 唐乾林 陶洪 顾正刚

康桂花 曹毅 眭碧霞 梁明

黄能耿 裴有柱

秘书长 胡毓坚

出版说明

根据《教育部关于以就业为导向深化高等职业教育改革的若干意见》中提出的高等职业院校必须把培养学生动手能力、实践能力和可持续发展能力放在突出的地位,促进学生技能的培养,以及教材内容要紧紧密结合生产实际,并注意及时跟踪先进技术的发展等指导精神,机械工业出版社组织全国近60所高等职业院校的骨干教师对在2001年出版的“面向21世纪高职高专系列教材”进行了全面的修订和增补,并更名为“全国高等职业教育规划教材”。

本系列教材是由高职高专计算机专业、电子技术专业和机电专业教材编委会分别会同各高职高专院校的一线骨干教师,针对相关专业的课程设置,融合教学中的实践经验,同时吸收高等职业教育改革的成果而编写完成的,具有“定位准确、注重能力、内容创新、结构合理和叙述通俗”的编写特色。在几年的教学实践中,本系列教材获得了较高的评价,并有多品种被评为普通高等教育“十一五”国家级规划教材。在修订和增补过程中,除了保持原有特色外,针对课程的不同性质采取了不同的优化措施。其中,核心基础课的教材在保持扎实的理论基础的同时,增加实训和习题;实践性较强的课程强调理论与实训紧密结合;涉及实用技术的课程则在教材中引入了最新的知识、技术、工艺和方法。同时,根据实际教学的需要对部分课程进行了整合。

归纳起来,本系列教材具有以下特点:

- 1) 围绕培养学生的职业技能这条主线来设计教材的结构、内容和形式。
- 2) 合理安排基础知识和实践知识的比例。基础知识以“必需、够用”为度,强调专业技术应用能力的训练,适当增加实训环节。
- 3) 符合高职学生的学习特点和认知规律。对基本理论和方法的论述要容易理解、清晰简洁,多用图表来表达信息;增加相关技术在生产中的应用实例,引导学生主动学习。
- 4) 教材内容紧随技术和经济的发展而更新,及时将新知识、新技术、新工艺和新案例等引入教材。同时注重吸收最新的教学理念,并积极支持新专业的教材建设。
- 5) 注重立体化教材建设。通过主教材、电子教案、配套素材光盘、实训指导和习题及解答等教学资源的有机结合,提高教学服务水平,为高素质技能型人才的培养创造良好的条件。

由于我国高等职业教育改革和发展的速度很快,加之我们的水平和经验有限,因此在教材的编写和出版过程中难免出现问题和错误。我们恳请使用这套教材的师生及时向我们反馈质量信息,以利于我们今后不断提高教材的出版质量,为广大师生提供更多、更适用的教材。

机械工业出版社

前 言

随着计算机软件的普及应用和迅猛发展,人们越来越关注软件的质量问题,就是被人们广泛使用的比较成熟的软件,也难免有不尽如人意之处,这是因为软件设计的缺陷、遗漏、错误带来的后果。由此,软件测试技术得到了人们普遍的关注。

本书共分为3部分7章。第1部分为软件测试基础篇,包括软件测试概述和测试设计与管理;第2部分为软件测试方法篇,包括生存周期测试方法、黑白灰盒测试方法、功能与性能测试方法;第3部分为现代测试技术篇,包括现代测试方法和自动测试方法。

本书以现代软件测试思想为指导,除比较全面地讲述传统软件测试技术和方法外,特别突出了生命周期软件测试概念、现代软件测试技术、主流测试工具以及典型测试方法的应用等,以帮助学生了解和掌握现代软件测试的原理、模型、方法和技术,并能选择合适的软件测试工具辅助进行相关的测试。针对软件开发方法和技术的发展变化、我国软件外包服务的蓬勃兴起以及软件高可靠性的要求,精心筛选内容以及重要的知识点,对支撑现代软件测试技术应用的测试方法进行了全面的介绍,力求反映现代软件测试技术发展的成果,为培养高素质、专业化的软件测试人才打下基础。

本书不仅注重内容的系统性、科学性,还特别关注测试技术的先进性、新颖性;编写时注重理论与实践紧密结合,实用性、可操作性强。

本书结合高职高专人才培养的需求,整体结构清晰,层次分明,逻辑性、针对性强;内容由浅入深,逐步深入,循序渐进,便于教师讲解、学生学习;大量的实训、习题紧密结合实际,便于提升学生的操作技能,调动学生学习的积极性、主动性,激发学生的想象力和创造力。

由于软件测试技术涉及的知识面广,不可能面面俱到,加上作者水平有限,书中难免有不当之处,请读者批评指正。

编 者

目 录

出版说明	2.2.2 测试计划制订	39
前言	2.2.3 测试报告	41
	2.3 实训	47
	2.4 习题	48
第1部分 软件测试基础篇		
第1章 软件测试概述	第2部分 软件测试方法篇	
1.1 软件缺陷与软件测试	第3章 生存周期测试方法	51
1.1.1 软件缺陷	3.1 生存周期文档审查	51
1.1.2 软件测试	3.1.1 生存周期文档审查概述	51
1.2 软件测试模型	3.1.2 可行性分析阶段	52
1.2.1 V模型	3.1.3 开发计划阶段	53
1.2.2 W模型	3.1.4 需求分析阶段	55
1.2.3 X模型	3.1.5 架构设计阶段	59
1.2.4 H模型	3.1.6 概要设计阶段	61
1.2.5 前置模型	3.1.7 详细设计阶段	66
1.3 软件测试方法及分类	3.1.8 系统测试阶段	67
1.3.1 按生存周期分类	3.1.9 系统交付阶段	69
1.3.2 按实施测试部门分类	3.2 传统测试阶段测试	71
1.3.3 按是否执行被测软件分类	3.2.1 代码会审	71
1.3.4 按是否检查程序代码分类	3.2.2 单元测试	77
1.3.5 按测试功能性能分类	3.2.3 集成测试	79
1.3.6 按是否使用自动测试工具分类	3.2.4 系统测试	81
1.3.7 按应用领域分类	3.3 验证阶段测试	84
1.3.8 其他分类方式	3.3.1 冒烟测试	84
1.4 软件测试的步骤	3.3.2 确认测试	85
1.5 习题	3.3.3 验收测试	87
第2章 测试设计与管理	3.3.4 用户测试	91
2.1 测试设计	3.3.5 独立测试	92
2.1.1 测试设计概述	3.4 交付阶段测试	93
2.1.2 测试需求分析	3.4.1 安装环境测试	93
2.1.3 测试用例	3.4.2 试运行测试	93
2.1.4 测试用例管理	3.4.3 安装/卸载测试	94
2.2 测试管理	3.5 运行维护阶段测试	95
2.2.1 测试管理概述	3.5.1 软件维护	95

3.5.2	回归测试	96	5.4	安全性测试	161
3.6	实训	97	5.4.1	安全性测试概述	161
3.7	习题	97	5.4.2	安全性测试方法	161
第4章	黑、白、灰盒测试方法	100	5.4.3	安全性测试过程	162
4.1	黑、白、灰盒测试方法概述	100	5.4.4	系统安全测试的内容	163
4.1.1	黑、白、灰盒测试方法简介	100	5.4.5	安全性测试用例设计	166
4.1.2	黑、白、灰盒测试的优缺点	101	5.5	Web 应用系统测试	168
4.1.3	黑、白、盒测试与静态测试的关系	103	5.5.1	Web 应用系统功能测试	168
4.2	黑盒测试	103	5.5.2	Web 应用系统性能测试	169
4.2.1	黑盒测试概述	103	5.5.3	Web 应用系统用户界面测试	170
4.2.2	黑盒测试方法	104	5.5.4	Web 应用系统客户端兼容性测试	172
4.2.3	黑盒测试用例设计	105	5.5.5	Web 应用系统安全测试	172
4.3	白盒测试	120	5.6	实训	173
4.3.1	白盒测试概述	120	5.7	习题	173
4.3.2	白盒测试方法	122			
4.3.3	静态结构分析法	122	第3部分	现代测试技术篇	
4.3.4	逻辑覆盖法	124	第6章	现代测试方法	175
4.3.5	基本路径法	128	6.1	面向对象测试方法	175
4.3.6	程序插桩法	130	6.1.1	面向对象测试概述	175
4.3.7	域测试	133	6.1.2	面向对象测试模型	178
4.3.8	符号测试	133	6.1.3	面向对象测试的方法	180
4.3.9	Z 路径覆盖	134	6.2	敏捷测试方法	185
4.3.10	程序变异	135	6.2.1	敏捷测试概述	185
4.4	灰盒测试	137	6.2.2	敏捷测试的流程	190
4.4.1	灰盒测试概述	137	6.2.3	敏捷测试的实施	192
4.4.2	灰盒测试应用	140	6.2.4	敏捷测试的管理	192
4.5	实训	141	6.3	探索性测试	194
4.6	习题	144	6.3.1	探索性测试概述	194
第5章	功能与性能测试方法	147	6.3.2	探索性测试的过程	197
5.1	功能与性能测试简介	147	6.3.3	探索性测试的管理	197
5.2	功能测试	148	6.4	模糊测试	198
5.2.1	功能测试概述	148	6.4.1	模糊测试概述	198
5.2.2	功能测试的实施	150	6.4.2	模糊测试的阶段	200
5.3	性能测试	154	6.4.3	网络协议模糊测试	201
5.3.1	性能测试概述	154	6.4.4	Web 应用程序模糊测试	202
5.3.2	几种主要的性能测试	155	6.5	云测试	202
5.3.3	性能测试的步骤	156	6.5.1	云计算简介	202
5.3.4	性能测试方案	159	6.5.2	云测试概述	203

6.5.3	云测试抽象模型	205	7.3.1	捕获和回放技术	225
6.5.4	云测试平台搭建	206	7.3.2	脚本技术	225
6.5.5	云测试关键技术	207	7.3.3	自动比较技术	227
6.6	国际化和本地化测试方法	207	7.3.4	脚本预处理技术	227
6.6.1	国际化和本地化测试概述	207	7.4	自动测试框架	227
6.6.2	国际化测试	208	7.4.1	自动测试框架概述	228
6.6.3	本地化测试	209	7.4.2	自动测试框架分类	229
6.7	基于组件、模型的测试方法	213	7.4.3	自动测试框架的因素	230
6.7.1	基于组件的测试方法	213	7.4.4	自动测试框架设计	230
6.7.2	基于模型的测试方法	216	7.5	自动测试工具	231
6.8	实训	219	7.5.1	自动测试工具分类	231
6.9	习题	220	7.5.2	常用自动测试工具简介	234
第7章	自动测试方法	221	7.5.3	测试工具的选择	243
7.1	自动测试概述	221	7.6	实训	244
7.2	自动测试准备工作	223	7.7	习题	245
7.2.1	选择自动测试级别	223	附录 A	软件测试方法简介	246
7.2.2	实施自动测试的条件	224	附录 B	英汉对照表	250
7.3	自动测试技术	225	参考文献	254	

第 1 部分 软件测试基础篇

第 1 章 软件测试概述

本章要点

- 软件缺陷
- 软件测试
- 软件测试模型
- 软件测试的分类
- 软件测试的步骤

1.1 软件缺陷与软件测试

1.1.1 软件缺陷

1. 软件缺陷定义

软件缺陷 (Defect), 常常又被叫做 Bug, 即计算机软件或程序中存在的某种破坏软件正常运行的问题、错误, 或者隐藏的功能缺陷。

缺陷的存在会导致软件产品在某种程度上不能满足用户的需要。一般来说, 符合下面 5 种情况之一的, 即可称为软件缺陷。

- 1) 软件未实现产品说明书中标明的全部功能。
- 2) 软件出现了产品说明书中指明的不该出现的错误。
- 3) 软件功能超出产品说明书指明的范围。
- 4) 产品说明书中虽未明确指出, 但未达到一般软件所具备的要求。
- 5) 软件测试员认为软件难以理解、不易使用、运行速度缓慢, 或者最终用户不满意。

2. 软件缺陷产生的原因

在软件开发的过程中, 产生软件缺陷是不可避免的。软件缺陷的产生主要是由软件产品的特点和开发过程所决定的。软件缺陷主要来自软件本身、开发团队、技术问题、项目管理等方面。

(1) 软件本身

1) 产品需求不清晰, 导致设计目标偏离客户的需求, 从而引起功能或产品特征上的缺陷。

2) 系统结构非常复杂, 无法设计出一个清晰的层次结构或组件结构, 结果导致意想不

到的问题或系统维护、扩充上的困难；即使设计成良好的面向对象的系统，由于对象、类过于繁杂，很难完成对各种对象、类相互作用的组合测试，从而隐藏着一些参数传递、方法调用、对象状态变化等方面的问题。

3) 对程序逻辑路径或数据范围的边界考虑不够周全，漏掉某些边界条件，造成容量或边界错误。

4) 对一些实时应用，要进行精心设计和处理，保证精确的时间同步，否则会因时间上不协调或不一致而带来种种问题。

5) 没有考虑系统崩溃后的自我恢复、数据的异地备份或灾难性恢复等问题，从而存在系统安全性、可靠性的隐患。

6) 系统运行环境复杂，由于用户使用的计算机环境千变万化，同时用户的各种操作方式或输入数据的方法不尽相同，因此，容易引起一些特定用户环境下的问题。此外，在系统实际应用时，由于数据量很大，因而会引起强度或负载问题。

7) 如果软件中采用了新技术，则可能涉及事先考虑不到的技术或系统兼容的问题。

(2) 开发团队

1) 开发人员进行系统需求分析时对用户的需求理解不透彻，或者和用户的沟通存在一些困难。

2) 不同阶段的开发人员对需求的理解不一致。例如，软件设计人员对需求分析的理解有时存在偏差，编程人员对系统设计说明书中的某些内容重视不够，或存在误解等。

3) 对于设计或编程上的一些假定条件，相关人员没有充分沟通。

4) 项目组成员技术水平参差不齐，新员工较多，或培训不够等原因也容易引起各种问题。

(3) 技术问题

1) 算法错误。在给定条件下没能得出正确或准确的结果。

2) 语法错误。对于编译性语言程序，编译器可以发现这类问题，但对于解释性语言程序，只能在测试运行时才能发现。

3) 计算和精度问题。计算的结果没有满足所需要的精度。

4) 系统结构不合理、算法选择不科学，造成系统性能低下。

5) 接口参数传递不匹配，导致模块集成时出现问题。

(4) 项目管理

1) 管理人员不重视产品质量方面的计划，忽视需求分析、评审、测试等环节，反而遗留更多的缺陷。

2) 开发周期短，需求分析、设计、编程、测试等各项工作不能完全按照定义好的流程进行，工作时间不够充分，结果造成产品不完整、不准确，错误较多。开发周期短，还会给开发人员造成过大的压力，从而引起一些人为的错误。

3) 开发流程不够完善，存在诸多的随意性和缺乏严谨的评审机制，容易产生问题。

4) 文档不完善，风险估计不足等。

1.1.2 软件测试

1. 软件测试的定义

软件测试是软件工程中的一个极其重要的环节，是项目开发不可缺少的一部分。

软件测试就是利用人工或者自动手段按照测试方案和流程对产品进行功能和性能测试的过程，其目的在于检验它是否满足规定的需求或弄清预期结果与实际结果之间的差别。它是识别开发的（中间或最终）软件（整体或部分）的正确度、完整度和质量的过程。

2. 软件测试的对象

软件的测试不仅仅是对程序的测试，软件的测试应贯穿于整个软件生命周期的始终。在软件定义阶段产生的可行性报告、项目实施计划、软件需求分析说明书或系统功能说明书、在软件开发阶段产生的概要设计说明书、详细设计说明书，以及源程序等都是软件测试的对象。

实际项目中最重要也是最复杂的阶段仍是对源程序的测试，也是通常大家所说的狭义的软件测试。

3. 软件测试的目的

1) 验证软件是否满足软件开发合同或项目开发计划、系统/子系统设计文档、软件需求分析说明书、软件设计说明书和软件产品说明书等规定的软件质量要求。

2) 通过测试，发现软件缺陷。

3) 为软件产品的质量测定和评价提供依据。

4. 软件测试的意义

软件测试的意义在于，保证发布的产品达到了一定的质量标准，并保证软件需求的无偏差实现。

5. 软件测试的原则

软件测试从不同的角度出发会派生出两种不同的测试原则：从用户的角度出发，就是希望通过软件测试能充分暴露软件中存在的问题和缺陷，从而考虑是否接受该产品；从开发者的角度出发，就是希望测试能表明软件产品不存在错误，已经正确地实现了用户的需求，确立人们对软件质量的信心。

一般情况下，测试原则是从用户和开发者双方的角度出发进行软件产品测试的，通过测试，可以为用户提供放心满意的产品，并对优秀的产品进行认证。

根据上述原则，软件测试需要从以下几点着手。

1) 开发者应当把“尽早和不断地测试”作为座右铭。

2) 程序员应该避免检查自己的程序，测试工作应该由独立的专业的软件测试机构来完成。

3) 设计测试用例时，应该同时考虑到合法的和不合法的输入以及各种边界条件，特殊情况下要制造极端状态和意外状态，比如网络异常中断、电源突然断电等情况。

4) 一定要注意测试中错误集中发生的现象，这和程序员的编程水平和习惯有很大的关系。

5) 对测试错误结果一定要有一个确认的过程。一般由 A 组人员测试出来的错误，一定要由 B 组人员来确认，严重的错误可以召开评审会进行讨论和分析。

6) 制定严格的测试计划，排除测试的随意性，并把测试时间安排得尽量宽松，不要期望在极短的时间内完成一个复杂的、高水平的测试。

7) 回归测试的关联性一定要引起充分的注意，修改一个错误却引起更多错误出现的情况并不少见。

8) 妥善保存一切测试过程文档，包括测试计划、测试用例、出错统计和最终分析报告，

为日后测试的重现及维护提供方便。

1.2 软件测试模型

软件测试模型，主要包括 V 模型、W 模型、X 模型、H 模型及前置模型，下面对这些模型作逐一介绍。

1.2.1 V 模型

V 模型最早是由 Paul Rook 在 20 世纪 80 年代后期提出的，旨在提高软件开发的效率和改善效果。V 模型中的过程从左到右，描述了基本的开发过程和测试行为，见图 1-1。V 模型的价值在于它非常明确地标明了测试开发过程中的各个级别，并且清楚地显示了这些测试阶段和开发过程中各阶段的对应关系。

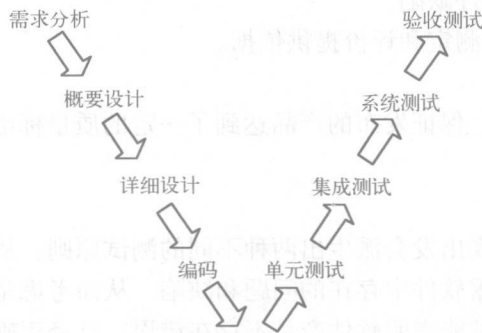


图 1-1 V 模型

V 模型指出，单元测试和集成测试应主要检测程序执行后是否满足软件设计的要求；系统测试主要检测系统功能、性能的质量特性是否达到系统要求的指标；验收测试则确定软件的实现是否满足用户的需要。

但 V 模型存在一定的局限性，它仅仅把测试作为在编码之后的一个阶段，是针对程序进行的寻找错误的行为，而忽视了测试行为对需求分析、系统设计等开发过程中应实现的功能进行验证和确认。

1.2.2 W 模型

W 模型，又称双 V 模型，记为 V&V，由 Evolutif 公司提出。相比 V 模型，W 模型增加了软件各开发阶段中应同步进行的验证和确认活动。W 模型由两个 V 字型模型组成，分别代表开发与测试两组过程，见图 1-2，图中明确表示出了开发与测试的并行关系。

W 模型强调，测试伴随于整个软件开发周期，而且测试的对象不仅仅是程序，需求、设计等也同样需要测试。也就是说，测试与开发应该是同步进行的。W 模型有利于尽早地全面地发现问题。例如，需求分析完成后，测试人员就应该参与到对需求的验证和确认活动中，以尽早地找出缺陷所在。同时，对需求的测试也有利于及时了解项目难度和测试风险，及早制定应对措施，这将显著减少总体测试时间，加快项目进度。

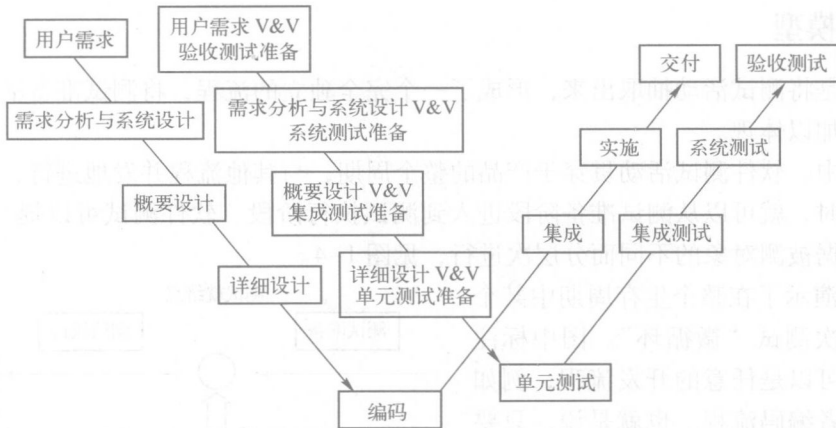


图 1-2 W 模型

但 W 模型也存在其局限性。在 W 模型中，需求、设计、编码等活动被视为串行的，同时，测试和开发活动也保持着一种线性的前后关系，上一阶段完全结束，才可正式开始下一阶段的工作。这样就无法支持含迭代关系的开发模型。对于当前软件开发复杂多变的现状，W 模型并不能消除测试管理面临的困惑。

1.2.3 X 模型

X 模型是由 Marick 提出的，目标是弥补 V 模型的一些缺陷，如交接以及经常性的集成等问题（见图 1-3）。

X 模型的左边，描述的是针对独立的程序片段所进行的彼此分离的编码和测试，然后再进行频繁的交接，通过编码集成最终合成为可执行的程序。模型的右上半部分，表示进一步对这些可执行程序进行测试。若已通过集成测试的成品则可以封版，并提交给用户，也可以作为更大规模集成的一部分。多根并行的曲线表示变更可以在各个部分发生。

X 模型还定位了探索性测试（右下半部分）。这是不进行事先计划的特殊类型的测试，诸如“我要这样测一下，结果会怎样？”，这一方式往往能帮助有经验的测试人员在测试计划之外发现更多的软件错误。



图 1-3 X 模型

由于 X 模型没有被文档化，其一开始的内容需要从 V 模型的相关内容中进行推断，以至于它还没有完全从文档上成为 V 模型的全面扩展。

V 模型和 W 模型均存在一些不妥之处。如前所述，它们都把软件的开发视为需求、设计、编码等一系列串行的活动，而事实上，这些活动在大部分时间内是可以交叉进行的，所以，相应的测试之间也并不存在严格的次序关系。

1.2.4 H模型

H模型是将测试活动抽取出来，形成了一个完全独立的流程，将测试准备活动和测试执行活动分别加以体现。

H模型中，软件测试活动贯穿于产品的整个周期，与其他流程并发地进行，当某个测试点准备就绪时，就可以从测试准备阶段进入到测试执行阶段。软件测试可以提早随时进行，并且可以根据被测对象的不同而分层次进行，见图1-4。

图1-4演示了在整个生存周期中某个层次上的一次测试“微循环”。图中标注的其他流程可以是任意的开发流程，例如设计流程或者编码流程。也就是说，只要测试条件成熟，测试准备活动完成后，测试执行活动就可以进行。

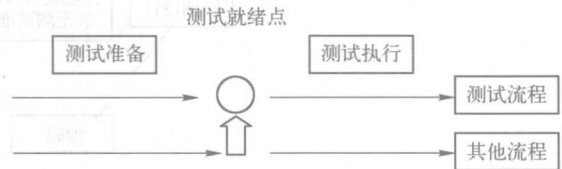


图1-4 H模型

H模型揭示了一个原理：软件测试是一个独立的流程，应贯穿产品整个生存周期，与其他流程并发地进行。H模型指出软件测试要尽早准备，尽早执行。不同的测试活动可以按照某个次序先后进行，但也可能是反复的，只要某个测试达到准备就绪点，测试执行活动就可以开展。

1.2.5 前置模型

前置模型由 Robin F Goldsmith 等人提出，是一个将测试和开发紧密结合的模型。该模型将开发和测试的生存周期整合在一起，随项目开发开始到结束，关系到每个阶段，每个关键行为，要求对每一个交付的内容都进行测试，见图1-5。

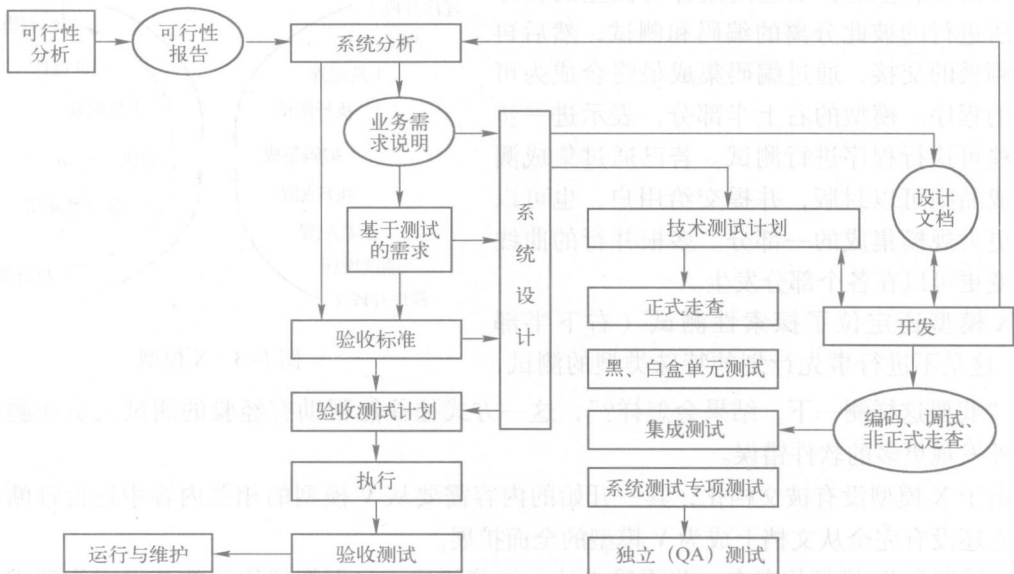


图1-5 前置模型

1.3 软件测试方法及分类

可以说，自从软件诞生之日起就随之出现了软件是否可靠的问题，也就是说，自从有了软件，软件的测试也就随之产生了。软件测试是一项非常复杂的系统工程，人们在实践中逐步摸索，总结了一系列行之有效的测试方法。

关于软件测试的方法，目前尚无一个统一的分类标准，从不同的角度考虑可以有不同的划分方法。为了更好地掌握软件的测试方法，更准确地了解各种测试方法的原理、流程、步骤及测试的过程，尽可能地对软件进行较全面的测试，有必要对软件的测试方法进行归类划分。本书整合了目前常用的一些软件测试分类方法，给出了软件测试的分类，如表 1-1 所示。

表 1-1 软件测试分类

分类方式	测试阶段/名称		主要测试方法
生存周期	定义阶段	可行性分析阶段测试	文档审查，评审论证
		开发计划阶段测试	文档审查，评审论证
		需求分析阶段测试	文档审查，审核分析
	开发阶段	架构设计阶段测试	文档审查，审核分析
		概要设计阶段测试	文档审查，审核分析
		详细设计阶段测试	文档审查，审核分析
		程序设计阶段测试	代码走查，调试排错
		传统测试阶段测试	代码会审、单元测试、集成测试、系统测试等
		验证阶段测试	冒烟测试、确认测试、验收测试、用户测试、独立测试等
	交付阶段测试	安装环境测试、试运行测试、安装/卸载测试等	
运行维护阶段	运行维护阶段测试	回归测试	
实施测试部门	开发方测试	综合运用各种测试方法	
	用户方测试	α 测试、 β 测试	
	第三方测试	国际、国家、行业、企业标准方法	
是否执行被测软件	静态测试	静态分析、手工测试、静态测试工具、黑盒测试等	
	动态测试	白盒测试、灰盒测试等各种测试方法	
是否检查程序代码	黑盒测试	等价类划分、边界值分析、因果图、随机测试、错误推测等	
	白盒测试	逻辑驱动、基路测试等	
	灰盒测试	黑盒测试法、白盒测试法	
测试的功能性能	功能测试	逻辑功能测试、易用性（界面）测试、易操作性测试等	
	性能测试	一般性能测试、安全性测试、稳定性（可靠性）测试、负载测试、压力测试、容错性（强度）测试、容量测试、易恢复性（恢复性）测试、可维护性测试、可移植性测试等	
是否使用自动测试工具	手工测试	静态分析等	
	自动测试	自动测试工具等	

(续)

分类方式	测试阶段/名称	主要测试方法
应用领域	国际化测试	国际标准方法
	本地化测试	本地标准方法
其他	通过测试	综合运用各种测试方法
	失败测试	综合运用各种测试方法
	基准测试	综合运用各种测试方法
	配置测试	硬件配置测试、软件配置测试

1.3.1 按生存周期分类

传统的软件测试是一个狭义的概念，仅仅是软件生存周期中的一个阶段，而现代的软件测试是一个广义的概念，不仅是软件开发过程中的一个重要阶段，而且是贯穿于软件生存周期的全过程，从软件开始诞生、软件开发完成、软件运行使用、软件维护升级、直至废止消亡的全过程。

下面，对软件生存周期各阶段的测试逐一作简单介绍。

1. 定义阶段

(1) 可行性分析阶段测试

测试时机：可行性分析报告提交后。

测试内容：可行性分析报告。

测试目的：项目是否可行、是否值得开发。

主要测试方法：文档审查，评审论证。

方法简介：开发公司组织由市场、开发、财务、质量等方面的专家在调研基础上对可行性分析报告进行审核分析。

(2) 开发计划阶段测试

测试时机：开发计划报告提交后。

测试内容：开发计划报告。

测试目的：项目计划是否可行、能否如期完成。

主要测试方法：文档审查，评审论证。

方法简介：开发公司组织由开发、质量和客户等方面的专家在调研基础上对开发计划报告进行审核分析。

(3) 需求分析阶段测试

测试时机：需求分析说明书提交后。

测试内容：需求分析报告。

测试目的：项目需求是否符合市场潮流，是否满足客户要求。

主要测试方法：文档审查，审核分析。

方法简介：开发公司组织由市场、开发、质量和客户等方面的专家在调研基础上对需求分析报告进行审核分析。

2. 开发阶段

(1) 架构设计阶段测试

测试时机：架构设计说明书提交后。